journal homepage: www.intl.elsevierhealth.com/journals/ijmi

Putting the technical back into socio-technical systems research

Enrico Coiera *

Centre for Health Informatics, University of New South Wales (UNSW), Sydney, NSW 2055, Australia

ARTICLE INFO

Article history:
Received 10 March 2006
Accepted 11 May 2006

Keywords:
Human–computer interaction
Information system design
Information system evaluation
Socio-technical systems

ABSTRACT

Socio-technical systems (STS) analysis has provided us with a powerful framework with which to analyse the reasons behind the poor acceptability, uptake and performance of many information or communication technology systems (ICT). However, for the contribution of STS thinking to be more than simply a means of critiquing current practices and ICT systems, it needs to also contribute to the process of developing new and more effective ICT systems. Specifically, we need to develop a formal design language for translating our insights about the socio-technical nature of work, into design specifications that result in better interventions in the work place. We need to get 'technical' about what we mean and about what we want from a design, and we need to work alongside technologists to shape technology, as well as the processes, organisations and cultures within which they will be embedded. Indeed the process of design itself can be seen as a socio-technical one, and understanding the decision to design itself may allow us one day to stop designing *for* people, and create STS that sustainably design themselves.

1. Introduction

Socio-technical systems (STS) science has arisen in response to the challenges of understanding complex technical systems that are embedded in a human world [1]. Even the most exquisitely designed technical artefact, if it comes past human hands, ends up performing in contexts undreamt of by its designers, and behaving in ways it was not meant to [2]. Sometimes the unexpected interactions between humans and artefacts produce wonderfully unanticipated serendipity, but often the unexpected outcome is a problem. Unanticipated errors, system failures, cost overruns and break-downs all can occur because a technology is used, or misused, beyond the strict constraints of its formal design specifications.

The socio-technical view attempts to understand the contribution of phenomena at the human social level to the performance of technical systems. Socio-technical systems analysis has provided us with a powerful framework with which to analyse the reasons behind the poor acceptability, uptake and performance of many information and communication technology (ICT) interventions [3]. Perhaps because health care systems are so dependent on complex human organisational structures, they seem particularly suitable to socio-technical analysis [4]. Information technologies seem crucial to the development of sustainable health services, but every IT intervention seems to generate an unanticipated consequence. It is with some concern that many now are recognising that the unanticipated consequences of IT use in health care include mishaps and errors that may have negative consequences on patient care [5].

Yet, perfect or not, information technology is no longer an optional extra. If health services in developed nations are to be sustainable over the next few decades, they will have to increase their efficiency, in the face of a decreasing workforce, an ageing population, and a diminished resource base [6]. It is clear that the types of system needed to do more

* Tel.: +61 2 9385 9026; fax: +61 2 9385 9006.
E-mail address: e.coiera@unsw.edu.au.

1386-5056/$ – see front matter © 2006 Published by Elsevier Ireland Ltd.
doi:10.1016/j.ijmedinf.2006.05.026

Table 1 – Health informatics: the sacred and the profane (a non-exhaustive comparison)

Sacred	Profane
The computer	Paper
The EMR	Politics
Terminologies	User complaints
System architectures	System implementation
Intelligent decision support technologies	System failures

with less will need to include elements of process automation, knowledge management and enhanced communication and interaction between health care workers. Individual health workers, unable to ever read or master the growing biomedical literature or 'bibliome', will be ever more reliant on contacting others with special expertise, or using intelligent computational systems to retrieve, synthesise and then act upon the distributed human knowledge base. Consumers inevitable will have a greater onus upon them to be more active and knowledgeable participants in the process of health care, and will thus have similar needs. Future health services and systems will also need to be 'designed' to be certifiably safe and efficient, unlike the present day, where the state of the art is a catalogue of avoidable deaths or misadventures for patients experiencing the unanticipated consequences of care in a complex unsafe system.

Much of the growing STS literature in health informatics is focussed at the phenomenological level, identifying the types of socio-technical interactions that occur when humans use IT in health settings. STS analysis is slowly revealing a hidden layer of phenomena that are wrapped around IT implementations, shedding light on why apparently well-intentioned system designs end up being rejected by their users, or why they under-perform or fail. Often STS thinking is used as a means of critique, pointing out the limitations of technology-driven thinking, when faced with the deep complexity of human social and organisational structures. STS research thus often eschews the sacred informatics ground of technology to examine the profane issues of people, failure and implementation (Table 1).

However, STS analysis can at its most extreme become a form of socio-ludditism, an anti-technology belief that because technology in human hands under-performs or misbehaves, it must be bad. Where once users were to blame when technology was not used the way it was designed, it now sometimes seems that technologists are to blame for not designing for all the ways in which their systems are misused.

Socio-technical systems thinking needs to be much more than simply a means of critiquing current practices and ICT systems. It needs to also contribute to the process of developing new, safer and more effective ICT systems. What is therefore needed is a way of describing events at the socio-technical level, connecting them to system behaviours and then to artefact design [7]. Specifically, we need to develop a formal design language for translating insights about the socio-technical nature of clinical work, into design specifications that result in better interventions in the work place. We need to get 'technical' about what we mean when we describe socio-technical events, 'technical' about what we want from a system design, and we need to work alongside technologists to shape technology, as well as the processes, organizations and cultures within which they will be embedded.

2. Getting technical about describing socio-technical processes

Technical systems have social consequences, and social systems have technical consequences [6]. Any descriptive logic we might choose to develop for socio-technical systems needs therefore to be able to describe social processes and events, as well as technical system behaviours and the way technical form and function are related to design. Our descriptive logic also needs to clearly describe how each articulates with the other.

One simple way to conceptualise this description space is as a series of layers [2], each embedded within the other (Fig. 1). Each layer has its own intrinsic properties, and built artefacts in each layer may have unexpected behaviours when influenced by events in the layers above it. Equally, phenomena in any one layer are always constrained by what has been engineered in the layers beneath:

1. *Algorithms*, which in essence are any formal representation of a process to do something, are put at the core. Algorithms might codify how a clinical process is executed, how a drug dose is calculated, or the protocol for deciding who can be consulted to answer a question. There are many formal methods for deciding if an algorithm is complete or correct, and these methods are the province of decision science, logic and mathematics.

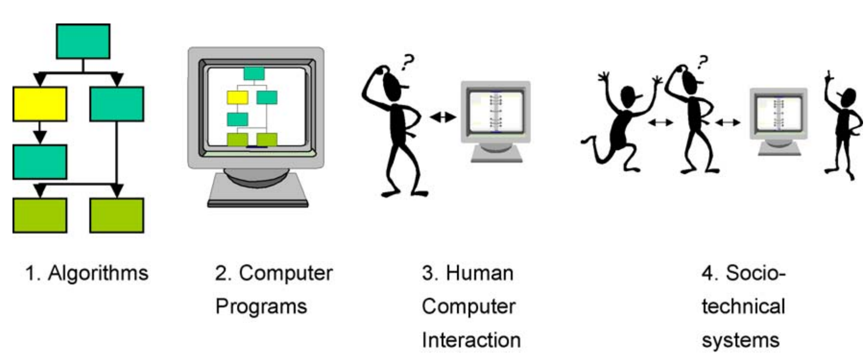

Fig. 1 – Four levels of information system design, with each level embedded in the following one (from Coiera, 2003).

2. *Computer programs* are a computational representation for our algorithms, so that the algorithm can be executed in a machine environment. Theoretical computer science provides us methods for deciding if the algorithm, when executed on a computer, will ever come up with an answer, how long that might take if it does, and different ways to design programs so that they are easy to write, to maintain, and to test that they are error free.
3. *Human–computer interaction (HCI)* theory provides us with the physical and metaphoric ways that an individual user can interact with a computer. Keyboards, mice, haptic surfaces and eye tracking are all physical ways that human intentions are transduced into something a computer understands, and icons and windows are examples of metaphors for actions and structures in the information space that are easily understood and operated by humans. HCI provides us with the means to study how people work, and how identified user needs can be translated into an *interaction design* that will describe how human and computer will work together. The science of HCI is cognitive science, which provides a rigorous framework for understanding how humans process information, and the resources like memory and attention (typically limited) that humans can bring to an interaction with a computer.
4. *Socio-technical systems* analysis focuses on the way *interactions between humans* restrict or shape *interactions between humans and technology*. People's behaviour is firstly shaped by the intrinsic social nature of the human animal. Research has clearly demonstrated how we apply the social rules used to govern human-to-human interactions, to our interactions with technology [8]. People's behaviour with technology is also clearly shaped by what is happening concurrently in the social domain. Our willingness to adopt a system, or adopt safe usage practices, or educate ourselves in system capabilities, is modified by the attitudes of others to these behaviours. Our capacity or willingness to allocate cognitive resources assumed by HCI design at level three is determined by who else we are currently interacting with at the social level.

If we adopt this four level framework, then we need only develop a descriptive logic for level four, and inherit the many existing frameworks from the previous three layers. It makes no sense, for example, to recreate the cognitive frameworks applied in HCI theory, but to simply adopt them. The framework also makes it clearer what is 'new' about STS studies (Table 2). It seems, for example, that occasionally what is presented as an STS analysis of clinical work-practices is really an HCI analysis. HCI researchers are focussed on the cognitive resources, limitations and capabilities of individuals, but STS researchers add in the extra dimension created when additional human agents shape the responses available to an individual using a system. A beautiful interaction design, shaped to fit the needs of individuals, may perform poorly when placed in a social setting where other agents interrupt, competing for attention, time, and scarce cognitive resource, or when cultural biases or social norms shape the responses and beliefs of individuals to the technology.

For example, the design of electronic health records (EHRs) typically focuses on the interaction between a single clinical user and the record that is being written or read, and the implicit goal is to optimise the process of information capture, and the re-use of that information in other systems for other purposes. But alternate conceptions are possible. For example, others populate the same interaction space as the user of an EHR. Indeed, in many settings the EHR user is often not the unique author, but is recording the result of a discussion with colleagues in their team. If the goal of the record system design is to obtain the highest quality information, then it may make much more sense to focus on events before the point of information capture, rather than focussing on improving the quality just at the capture stage. For example, it may be more important to support the collaborative discussion between clinicians than to over-engineer the act of record transcription into the system. Similarly, electronic prescribing systems have been shown to over-engineer the interaction between prescriber and computer, and ignore the collaborative discussions that would ensure that the prescriber has made an appropriate decision [9,10]. Forcing doctors to prescribe at a computer terminal, disconnected from their colleagues, by definition circumvents the traditional bed-side discussions between doctor and nurse that often provide crucial information that informs the prescription of medication.

Another clear but daunting message from this analysis is the broad number of other disciplines that we could or should be aware of when considering socio-technical designs, and also reminds us to avoid reinventing them. The other much more exhilarating message is that, whilst levels one through to three are all now well established disciplines in their own rights, the socio-technical space is still relatively unexplored, and ripe for research. If there is going to be invention and discovery is health system design that has broad value to others beyond the world of healthcare, then it may very well come from the socio-technical science we develop in health to serve our own needs.

Table 2 – Attributes of socio-technical systems (STS) design compared to standard human–computer interaction (HCI) design considerations

HCI	STS
Allocation of agent resources for a transaction, e.g. working memory, attention, awareness, time, ...	Competition for agent resources by external agents and transactions (cognitive, task priority, time, ...)
Decision biases and beliefs	Cultural biases and social norms
Knowledge gaps internal to agent	Knowledge gaps between agents (common ground)

3. Getting technical about socio-technical system design

When shaping artefacts that are to be used in organisational settings, we do not just design technology, we have to design entire socio-technical systems [6]. In other words, the objects of our design are not just specific protocols, or technologies, or even individual human–technological interactions, but include an attempt to shape, or at least accommodate, the social environment.

Design and evaluation are entwined processes, which, respectively, shape artefact design or requirements for design (Fig. 2). Evaluation methods are now usually understood to be either *formative*, exploring the kinds of needs that might exist, or *summative*, testing whether these hypothesised needs have been met or need to be revised [11]. Similarly we might describe designs as either being *provocative*, used as a means to generate responses from people to illuminate their needs, or *summative*, attempting to meet the needs that have been understood.

Everything is designed to meet some purpose, within some context [2]. Theories of context thus shape what we choose to account for in design and testing. One potentially powerful theoretical approach is to see organisational context as a multi-agent universe, with interactions between agents mediated by technologies, and commitments to particular tasks, agents or tools are shaped by scarcity of resources, and the resultant competition for them [7]. With this theory of context, evaluation of the performance of a STS only makes sense if there is an account given of the resources that are available in the environment and the competition that occurred for those resources at the time of the evaluation. For example, one would not attempt to measure how long it takes to carry out a task using a piece of software in a clinical setting without knowing whether the person using the software was also busy with other tasks at the time, and whether they might have been interrupted by other people, or indeed other software elements, competing for their attention, memory and time. Seeing context shaped by scarcity of resources thus allows us to conceive of design goals targeted towards system effectiveness and safety, but should also be rich enough to handle design goals like sustainability, flexibility and adaptability.

Consequently our logic of design must take account of context, and specifically resources and competition for resources. Designing just for an individual using a system implicitly makes an assumption that the individual is completely committed to the task of using our system. This *single-task assumption* might make sense 'in vitro' when developing a design in controlled or closed-system test settings, but breaks down 'in vivo' when our informatics interventions are used in complex or open system organisational settings.

In an open environment, the commitment of agents to a task is always conditional. The process of STS design is thus challenging because we must make assumptions about resources, competition and commitment of agents, and these assumptions inevitably shape a design's purpose. Several different design strategies seem possible to accommodate these uncertainties:

- *Maximising*: Designs might take a normative strategy, making optimistic or best-case assumptions about resources, because there is no real downside if resources get taken away for a while, or if things are not as they 'should be'. A word processing package does not need to assume the complete attention of its user, and if a user is involved in multiple tasks that divert them from writing the document, the application's purpose is not critically subverted. On the other hand, if document creation is time or mission critical, then other resource assumptions might need to be factored into the design.
- *Satisficing*: Designers might want to recognise that resources levels may vary from time to time, and so shape their design

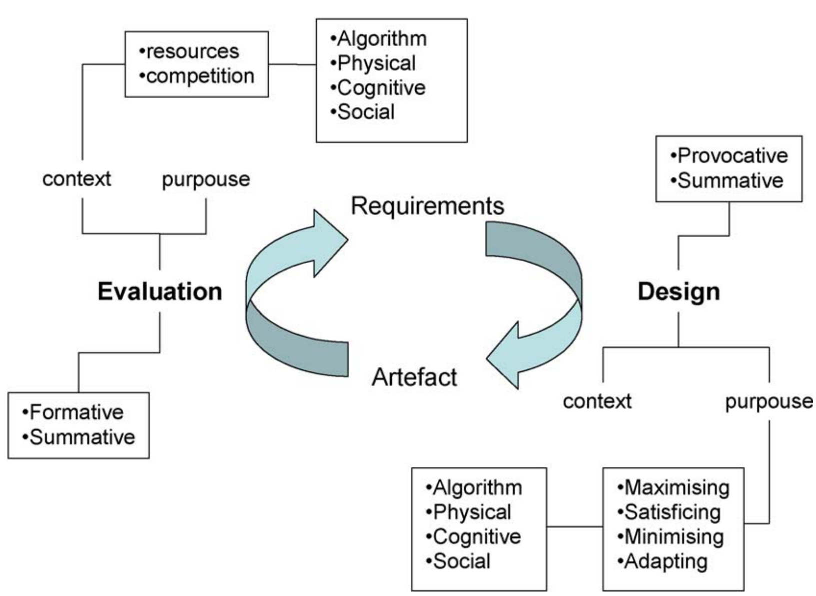

Fig. 2 – Evaluation and design shape the framing of requirements and artefacts, at each level of design from algorithmic description of task needs through to social setting.

to perform satisfactorily on average. Accounting for the ebb and flow of resources over a population of people or series of interactions requires some assessment of the typical work context, and an acceptance that sometimes, system performance will be less than ideal. The designer of an electronic prescribing package will not know exactly how much time each doctor has to use their system, but by measuring typical times that satisfy the needs of most users, they are likely to get their design right for most occasions.

- *Minimising*: The design may make pessimistic or worst-case assumptions, so system performance is never compromised if resources are taken away for a period. In safety-critical settings, this may be the best choice. For example, the design of an alert system that passes on life-critical information would not assume that just because the alert had been generated that it had been noticed. The designer might want to assume that messages are not always noticed, and incorporate a feedback look where the reader acknowledges message receipt, and willingness to act on the message. Failure to get receipt in a specified time might generate a new alert, perhaps using different channels, or be sent to different agents. The design of safety-critical systems has received much attention in other disciplines, and is increasingly of interest to health informaticians [12].
- *Adapting*: The most challenging, but also most flexible design will attempt to adapt to changes in its environment. Whether there is an attempt to embed intelligence in system response though environmental feedback measures, or an attempt to allow users to select operational modes depending upon their own assessment of their context, adaptive design have an extended operating range, and work across a number of different contexts. For example, a data display may provide many details of a patient's status on the assumption it has the full attention of the clinician using it, or it might condense the data into a very few critical task specific indicators, knowing that an emergency is being dealt with, and that the user is engaged in tasks which take them away from the system. In such circumstances, keeping a user 'in-the-loop' with changes to the data over the last few moments might become critical, to enhance situational awareness. Such changes to a display could easily be manually selected by a user, but could also be triggered by sensing data coming from patient monitoring equipment in settings like Intensive Care or surgery.

Just like design, the process of STS evaluation is shaped by context and purpose. Many technology acceptance models are targeted at the level of individual interactions, and try and assess the 'fit' between task, individual and technology to explain why some technologies work or fail in given settings [13]. However, evaluation also needs to account for the socio-technical assumptions about context implicit in the design [14]. An evaluation of a safety-critical system should thus test the context to see whether assumptions made in the design about likelihood of agent commitment, resources and competition are reflected in the environment being evaluated. Indeed, with a clearly articulated STS framework underpinning design, it might be possible to walk into a new environment, evaluate only the environment, and make a recommendation about which type of intervention will fit best, or generate appropriate design specifications. In complex organsiations, simulation tools offer great promise as they allow complex emergent interactions between agents to anticipated, and system performance to be explored under a variety of different resource and activity conditions.

4. Meta-design theories may lead us to adaptive socio-technical systems

So far, design has been described as a purposive act, where designers external to an environment are given some task to achieve, and set about understanding the environment and shaping an artefact until the design purpose is met. Nothing is said about who decides what is a 'need' or the type of solution that is best suited to that need. Yet design itself is a social process—what is chosen to be constructed or not constructed, accepted or rejected, is ultimately decided by people and their social processes [15]. Consequently, there is also scope for us to contemplate socio-technical theories *about* design, or meta-design.

One avenue for research that looks promising is to see design, not as a remote act external to the system, but as an emergent property of a system. The capacity for people to develop 'work arounds' has long been remarked upon, where people make technology do things that it was not intended to, or make it work in settings never intended. A Post-it note glued to a computer terminal, a hand-written alteration to a printed form, or local variations in protocols that are never written down but implicitly understood by locals, are all forms of work-arounds. People often will simply 'make-do' with these work arounds rather than explicitly articulate a need. Consequently work-arounds are implicit statements of need, and studying them should provide a powerful way of revealing the strengths and weaknesses of existing processes and systems, and identifying which needs need to be addressed with new interventions.

Work-arounds probably only happen when there is both pressing need to make something work that is not, and some latitude in the work place to evolve them, providing some degree of freedom to generate novel behaviours or technical modifications. Studying how work-arounds develop will probably show us that they evolve over time, with agents shaping their environment, and the altered environment then shaping agent behaviour. They will probably also show that agents will collaborate or interact in new ways that offer improvements over their current situation. Measuring the effort expended in working around might tell us a lot about how important then unmet need is, because people will be expending scarce resources to do it.

Whenever new tools are 'released' into an open system like an organization, we are thus likely to see new systems of behaviour evolve that better meet the real needs of the organization than the initial intervention design. Indeed, it is likely that most complex socio-technical systems are inherently self-designing, developing new behaviours and tools to meet the goals of the whole system. Studying the period of accommodation, from first introduction of a new intervention, to the point of equilibrium when humans have settled on the way they really want to use it, will say much about the real needs of an organization, and about the resources available to meet these needs. Consequently, rather than simply trying to evaluate how well people and technologies 'fit' when we put them together, we may get richer insights if we study *how they try to fit* together, and if we recognise more generally that in a socio-technical system, agents and environment co-evolve around each other, until they do fit.

Indeed, one might remark that historically information systems have been rejected because they have been designed somewhere external to the socio-technical system, and one metric to predict success of any new intervention is the degree to which the design process has been internal to the target system. Modern methods of 'participatory design', for example, are still often one step removed from this auto-design processes, as people internal to the system are invited 'outside' to help outsiders build something they will then send back 'inside'.

Finally, since people will seek novel solutions to their own problems where there is freedom to experiment and spare resource to innovate, then we might wonder how we can design truly sustainable organisational systems, which have the capacity to adapt themselves to their evolving needs. The health system at present is one that consumes enormous resource, and generates enormous waste, and would not meet any criterion of sustainability. Injecting new interventions from 'outside' the system, as we currently do in health informatics, is itself not a sustainable approach, as the capacity for external designers to meet all the evolving needs of those inside will just never be there. Could we use socio-technical constructs to understand the process of design from within the system, to design a system that more or less then designs itself?

5. Conclusion

The socio-technical view is not just the arrival of sociological thinking to explain what people do when they use technology. Socio-technical thinking allows us to think about a broad class of phenomena that are crucial to deciding which technology gets built, how it is designed, and how it fares once it is used in the real world. Formal theories of design that incorporate measurable socio-technical phenomena have the power to permit radical reconceptions about what we build, and how we build it. Indeed, if we allow ourselves to see the act of design itself as a socio-technical process, we begin to realise that the process of design can be re-imagined. Perhaps one day health informaticians will not design 'interventions' that are injected into a system, but orchestrate mechanisms that manufacture an infinite variety of 'accommodations', each one a self-fitting and emergent property of different locales. One day we may stop designing technology *for* people, and build socio-technical systems that design themselves.

REFERENCES

[1] E.L. Trist, The evolution of sociotechnical systems as a conceptual framework and as an action research program, in: A.H. van de Ven, W.F. Joyce (Eds.), Perspectives on Organization Design and Behavior, Wiley-Interscience, New York, 1981, pp. 19–75.

[2] E. Coiera, Guide to Health Informatics, second ed., Arnold, London, 2003.

[3] R.P. Bostrom, J.S. Heinen, MIS problems and failures: a socio-technical perspective. Part II: the application of socio-technical theory, Manage. Inf. Syst. Q. 1 (4) (1977) 11–28.

[4] M. Berg (Ed.), Health Information Management—Integrating Information Technology in Health Care Work, Routledge, London, 2004.

[5] J.S. Ash, M. Berg, E. Coiera, Some unintended consequences of information technology in health care: the nature of patient care information system related errors, J. Am. Med. Inform. Assoc. 11 (2004) 104–112.

[6] E. Coiera, Four rules for the reinvention of healthcare, Br. Med. J. 328 (2004) 1197–1199.

[7] E. Coiera, Interaction design theory, Int. J. Med. Inform. 69 (2003) 205–222.

[8] B. Reeves, C. Nass, The Media Equation, Cambridge University Press, Cambridge, 1996.

[9] M. Beuscart-Zéphir, S. Pelayo, P. Degoulet, et al., A usability study of CPOE's medication administration functions: impact on physician–nurse cooperation, Medinfo 11 (2004) 1018–1022.

[10] J. Aarts, M. Berg, A tale of two cities: a sociotechnical appraisal of the introduction of computerized physician order entry in two Dutch hospitals, Medinfo 11 (2004) 999–1002.

[11] C.P. Friedman, J.C. Wyatt, Evaluation Methods in Medical Informatics, second ed., Springer-Verlag, New York, 2006.

[12] J. Fox, S. Das, Safe and Sound: Artificial Intelligence in Hazardous Applications, The MIT Press, Cambridge (MA), 2000.

[13] M.T. Dishaw, D.M. Strong, Extending the technology acceptance model with task-technology fit constructs, Inf. Manage. 36 (1999) 9–24.

[14] V. Venkatesh, M.G. Morris, G.B. Davis, F.D. Davis, User acceptance of information technology: toward a unified view, Manage. Inf. Syst. Q. 27 (2003) 425–478.

[15] E. Coiera, Editorial: the impact of culture on technology, Med. J. Aust. 171 (1999) 508–509.